

Computational Fluid Dynamics

If you want to learn a bit more of the math behind fluid dynamics, read my previous post about the Navier-Stokes equations and Newtonian fluids. The equations derived in the post are the equations which govern motion of common fluids such as water, honey, and so on.

If you just want to see running code, it's on GitHub.

Simulations

Due to the complex nature of the Navier-Stokes equations, analytical solutions range from difficult to practically impossible, so the most common way to make use of the Navier-Stokes equations is through simulation and approximation. A number of simulation methods exist, and in the next section, we will examine one of the algorithms often used in computer graphics and interactive applications.

Smoothed Particle Hydrodynamics

One of the most researched and most used methods for fluid simulations, especially in interactive applications, is a particle-based approach known as Smoothed Particle Hydrodynamics (SPH). SPH is, at the most basic level, an attempt to discretize the Navier-Stokes equations. In addition to the fluid particles themselves, a smoothing function (called the smoothing kernel) is used which spreads the properties of each particle over a continuum. In fact, SPH is often called “an interpolation method for particle systems.” We will now derive and demonstrate the use of SPH for modeling a simple incompressible Newtonian fluid.

Smoothing Kernels and Basis of SPH

Since SPH is essentially an interpolation method for particle systems, one of the key ideas behind it is the ability to evaluate a scalar quantity in any point in space, even though the quantity only truly has a value at the discrete particles.

In order to evaluate a quantity A at a point \vec{r} , we use the following equation:

$$A(\vec{r}) = \sum_j m_j \frac{A_j}{\rho_j} \cdot W(|\vec{r} - \vec{r}_j|, h).$$

This is a weighted summation of all the particles around the point. A_j is the value of A at the particle, m_j is the mass of the particle, ρ_j is the density of the particle, and $W(|\vec{r} - \vec{r}_j|, h)$ is the weighting factor, and h is the maximum distance. The function W is also called the *smoothing kernel*, because it smoothes the particle values throughout continuous space.

The smoothing function W has a few notable properties. First of all, the function should decrease as the distance increases; this is only logical, since the points nearby should have more influence than those far away. Second of all, the kernel should evaluate to 0 at distances equal to or greater than h . Finally, the smoothing kernel should be normalized. A normalized smoothing kernel satisfies the equation

$$\int W(|\vec{r}|, h) d\vec{r} = 1.$$

Derivatives of Quantities

The Navier-Stokes equations often require calculating the first or second derivatives of various quantities. This is particularly easy in SPH, because the derivative of a quantity is

$$\frac{\partial}{\partial \vec{r}} A(\vec{r}) = \frac{\partial}{\partial \vec{r}} \sum_j m_j \frac{A_j}{\rho_j} \cdot W(|\vec{r} - \vec{r}_j|, h).$$

Equivalently,

$$\frac{\partial}{\partial \vec{r}} A(\vec{r}) = \sum_j m_j \frac{A_j}{\rho_j} \cdot \frac{\partial}{\partial \vec{r}} W(|\vec{r} - \vec{r}_j|, h).$$

Thus,

$$\nabla A(\vec{r}) = \sum_j m_j \frac{A_j}{\rho_j} \cdot \nabla W(|\vec{r} - \vec{r}_j|, h)$$

and

$$\nabla^2 A(\vec{r}) = \sum_j m_j \frac{A_j}{\rho_j} \cdot \nabla^2 W(|\vec{r} - \vec{r}_j|, h).$$

Calculating Density

As you may have noticed, the equation for the SPH value of a scalar quantity includes both a mass and a density term. This is because each particle represents a small volume $V_i = \frac{m_i}{\rho_i}$. The Navier-Stokes equations for an incompressible fluid (such as water) require that the density term is constant; however, due to the nature of SPH, maintaining a constant density is very difficult. Since in many applications accuracy is less important than realism and speed, we ignore the requirement that density is constant. We can later compensate for that by making the density restoring force (pressure) very strong.

Since density ρ is non-constant, we may need to calculate density at a given position. Since density is yet another intensive property, we can apply the SPH equation above to get

$$\rho_j = \sum_j m_j \frac{\rho_j}{\rho_j} W(|\vec{r} - \vec{r}_j|, h) = \sum_j m_j W(|\vec{r} - \vec{r}_j|, h).$$

The problem of density is one of the biggest reasons not to use SPH for models which require precision or accuracy. SPH is meant to provide a realistic-looking fluid simulation, at the cost of ignoring certain requirements, such as symmetry, conservation of momentum, or constant density.

Simulation

The Navier-Stokes equations can now be applied to the SPH particles. We have two main equations. The first is the conservation of mass equation,

$$\nabla \cdot \vec{v} = 0.$$

The second equation stems from the conservation of momentum, and states that

$$\rho \frac{Dv}{Dt} = -\nabla p + \mu \nabla^2 v + f.$$

The first equation is guaranteed for particle systems, because we have a constant amount of particles, each with a constant mass. Therefore, the total mass remains constant, and we can ignore the first equation. The second equation dictates the motion of the particles. The left hand term is the change in momentum. Therefore, we can express the acceleration of each particle as

$$a_i = \frac{f_i}{\rho_i}$$

where f_i is the sum of all the force terms on the right hand side. In order to be able to simulate the particle motion, we must express each of the force terms through a discrete summation.

Force Terms: Pressure

If we apply our SPH equation to pressure, we arrive at the equation

$$f_i^{pressure} = -\nabla p(\vec{r}) = -\sum_j m_j \frac{p_j}{\rho_j} \nabla W(|\vec{r}_i - \vec{r}_j|, h).$$

However, this isn't symmetric. Pressure should be symmetric, since every action should have an equal and opposite reaction; however, if we examine a 2-particle system, the pressure of particle i depends on the pressure of particle j and vice versa, so symmetry is not guaranteed in the general case. In order to rectify this, we can use a different pressure equation, which uses an average of particle pressures to ensure symmetry:

$$f_i^{pressure} = -\nabla p(\vec{r}) = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(|\vec{r}_i - \vec{r}_j|, h).$$

We must also remember that the only state a particle keeps around is its mass, density, and velocity. Thus, we cannot just use pressure, we must calculate it. Using the ideal gas state equation, we can say that

$$p = k\rho.$$

In order to make the simulation more numerically stable, we can make pressure smaller by using the difference. Since the equation uses the gradient of pressure, the math is unchanged if we use the equation

$$p = k(\rho - \rho_o)$$

where ρ_o is the rest density. By doing this substitution, we reduce the value of p , and thus decrease the effects of floating point rounding errors introduced by the computations.

Force Terms: Viscosity

Applying SPH principles to viscosity, we arrive at the equation

$$f_i^{viscosity} = \mu \nabla^2 v(\vec{r}_i) = \mu \sum m_j \frac{v_j}{\rho_j} \nabla^2 W(|\vec{r}_i - \vec{r}_j|, h).$$

However, we encounter the same problem as before, namely that the viscosity force is non-symmetric. We can resolve it in a similar manner, using the difference in velocities instead. The resulting equation takes the form of

$$f_i^{viscosity} = \mu \sum m_j \frac{v_j - v_i}{\rho_j} \nabla^2 W(|\vec{r}_i - \vec{r}_j|, h).$$

In physical terms, this equation states that a particle is accelerated in the direction that its environment is moving. Thinking about a floating particle in a stream of liquid, this makes sense.

Force Terms: Surface Tension

In a fluid, molecules experience intramolecular attraction forces. For a particle embedded inside the fluid, the force is evened out; however, for particles on the edge of the fluid, the force does cause an acceleration, and thus is non-negligible. Since each particle is the cause of the intramolecular attraction, we define another quantity (called “color” for convenience) such that c is equal to 1 at every particle and 0 otherwise. Therefore, the equation that describes the SPH quantity for c would be

$$c(\vec{r}) = \sum m_j \frac{c_j}{\rho_j} W(|\vec{r}_i - \vec{r}_j|, h).$$

Since c is 1 at all particle locations, we can simplify this to say that

$$c(\vec{r}) = \sum m_j \frac{1}{\rho_j} W(|\vec{r}_i - \vec{r}_j|, h).$$

For physical reasons that are beyond the scope of this analysis, the surface tension force is equal to

$$f_{surface} = \sigma \kappa \vec{n} = -\sigma \nabla^2 c \frac{\vec{n}}{|\vec{n}|}.$$

We try to avoid calculating the force when $|\vec{n}|$ is small, because that causes problems because we are dividing by a small floating point number. We only calculate the surface tension force when $|\vec{n}|$ is greater than a given preset. The final equation for the surface tension force (when we calculate it) would be

$$f_i^{surface} = -\sigma \nabla^2 c \frac{\nabla c}{|\nabla c|}.$$

Force Terms: External Forces

External forces are simply applied to every particle as they would be to a normal mass. If a particle touches (or goes into) a solid object, it is reflected (pushed out with it’s velocity vector reflected over the object normal vector).

Smoothing Kernels

The smoothing kernel functions are required to do the actual computations for simulation. The quality of simulation, as well as the speed and accuracy, highly depend on the choice of smoothing kernels. In the original SPH publication that investigated simulating fluids in real-time [2], the authors suggested the use of three different kernels for different things. Note that the smoothing kernels are piece-wise functions; the functions presented subsequently define the values of W for $0 \leq r \leq h$. In other cases, $W = 0$. Note that the following smoothing kernels are not normalized, and should be prior to use.

General Smoothing Kernel

The general smoothing kernel, used in all but two cases, is a sixth-degree polynomial with the equation

$$W_g(r, h) = \frac{315}{64\pi h^9} (h^2 - r^2)^3.$$

One of the benefits of this empirically designed kernel is that it uses r^2 , which means that we never need to calculate r . Calculating r would be an expensive operation, because it involves a square root; calculating r^2 , however, involves only multiplication, and is thus much faster.

Pressure Kernel

For calculations involving pressure, the general kernel becomes zero when particles are very close together. This is not what fluids actually act like - in fact, as particles come closer together, they should repel more due to pressure instead of less. Thus, Desbrun [6] suggests a different smoothing kernel of the form

$$W_p(r, h) = \frac{15}{\pi h^6} (h - r)^3.$$

Viscosity Kernel

The final smoothing kernel we are using is specific to the viscosity term. If we were to use the general smoothing term for viscosity, two particles which are very close to each other could possibly increase in speed. This is incorrect, since viscosity is a friction force, which should not be increasing the energy in the system. Thus, the authors of the original fluid simulation SPH paper suggest using the kernel of

$$W_v(r, h) = -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1.$$

Ongoing Research

Although the Navier-Stokes equations were originally formulated nearly 200 years ago, fluid mechanics is a rapidly evolving field. The advent of computer technology has allowed physicists to accurately model the behaviour of fluids through iterative approximations of their behaviour, making the study of the Navier-Stokes equations still more important. Current research can be subdivided into two categories: one form of research is theoretical, seeking mathematical proof of the existence and smoothness of the solution to Navier-Stokes; the other branch of ongoing research is more focused on practical application of Navier-Stokes to real-world problems. Though in some cases the distinction is purely artificial, it is nonetheless useful in discussing the various topics currently being investigated.

Theoretical Research

The fundamental theoretical problem of Navier-Stokes is one of the seven Millenium Prize Problems. The Millenium Prize Problems are seven mathematical problems put forth by the Clay Mathematics Institute in the year 2000; the first person to offer a valid solution to any of the problems is promised a million-dollar reward. The problem of Navier-Stokes, as stated by the Clay Mathematics Institute, is as follows:

“Prove or give a counter-example of the following statement:

In three space dimensions and time, given an initial velocity field, there exists a vector velocity and a scalar pressure field, which are both smooth and globally defined, that solve the Navier-Stokes equations.”

Though this statement itself has not been proven or indicated false, attempts at proof have come close. It is known that for Navier-Stokes in two dimensions, the solutions exist and are smooth. In addition, it is known that for low initial velocities, smooth solutions to 3-dimensional Navier-Stokes exist; however, for higher initial velocities, the existence of smooth solutions has not been proven.

Practical Research

Though the mathematical problem posed by Navier-Stokes has not yet been resolved, computer-aided approximations are suitable for many purposes. From the design of aerodynamic aircraft to simulation of water for Pixar films, fluid simulations are highly used in industry. Much current research in the field of computational

fluid dynamics (CFD) attempts to find faster or more-accurate approximation algorithms in order to make these simulations more useful still.

The methods used in CFD can be broken down into two categories: discretization models and turbulence models. Discretization models attempt to break the fluid into a discrete set of elements, and iteratively simulate the motion of the fluid through time using the Navier-Stokes equations. Turbulence models deal with turbulence, a chaotic type of fluid flow, which is nearly impossible to simulate as with the discretization models; instead, turbulence models attempt to track or obtain the value of specific properties of a fluid over time.

Though the basic equations used are the same, approximation methods often take different approaches, depending on their objective. Some methods, especially ones aimed towards very fast or interactive fluid simulation, use particle based modeling. A volume of fluid is subdivided into a finite number of particles, and each particle is subject to the velocity fields dictated by the Navier-Stokes equations as well as external forces. This model typically simplifies computation and implementation, and is commonly used in computer graphics and computer games. The particle nature is usually hidden by interpolating particle positions through a process called point splatting. Smoothed Particle Hydrodynamics, the method investigated in the previous section, falls under that category. Other methods, however, use a three-dimensional grid (a “mesh”) to model the fluid. The internals and boundaries are tracked through iterative Navier-Stokes approximations; the benefit of this method is a higher precision and higher flexibility in terms of interaction with other bodies, at the cost of computational speed.